

FIG. 1

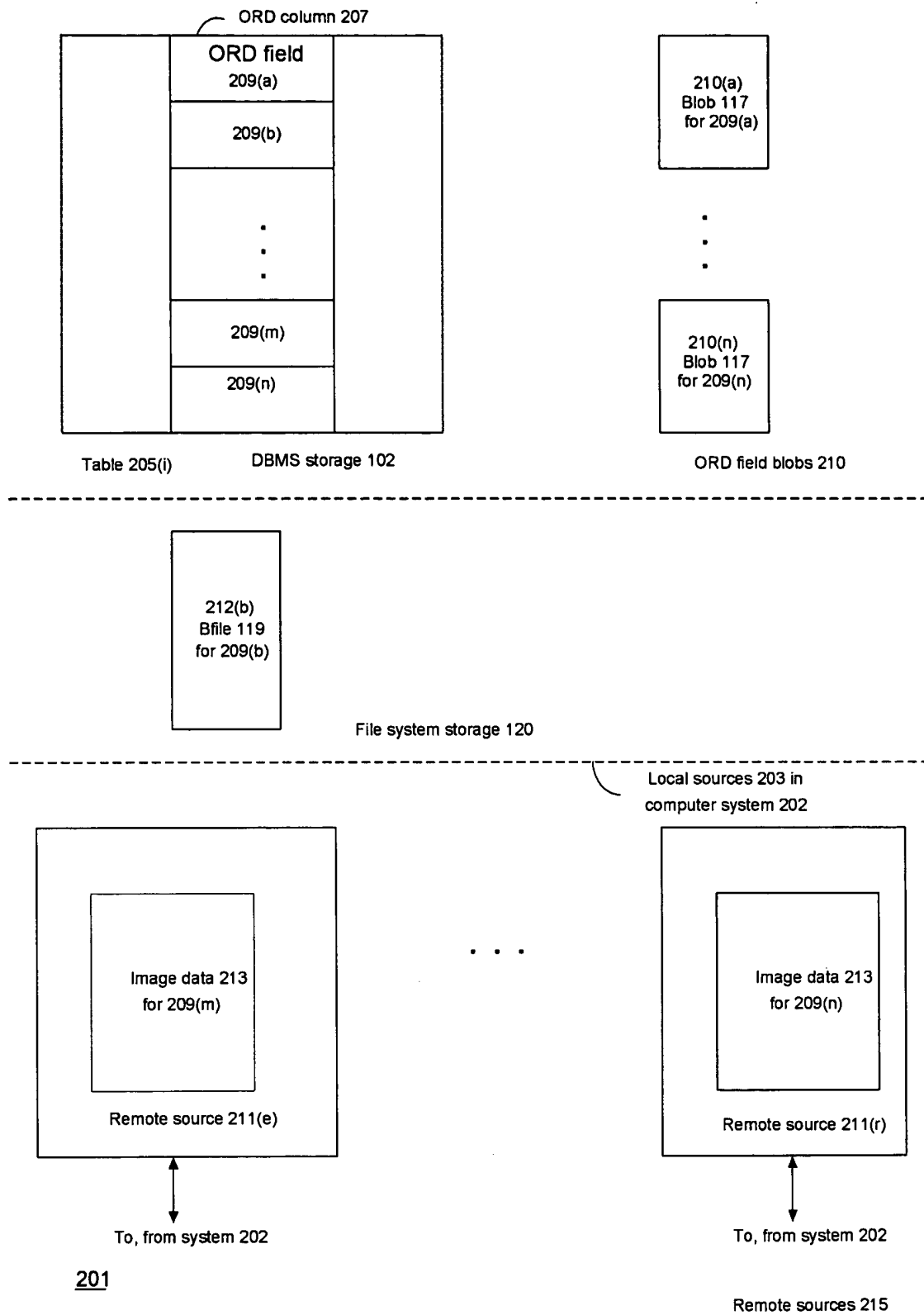


FIG. 2

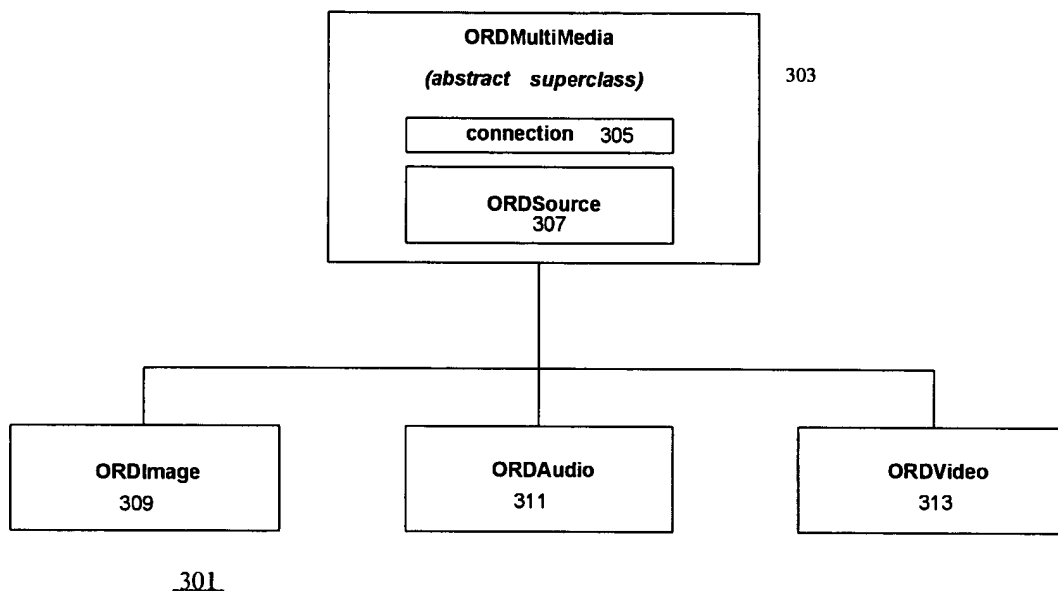


FIG. 3

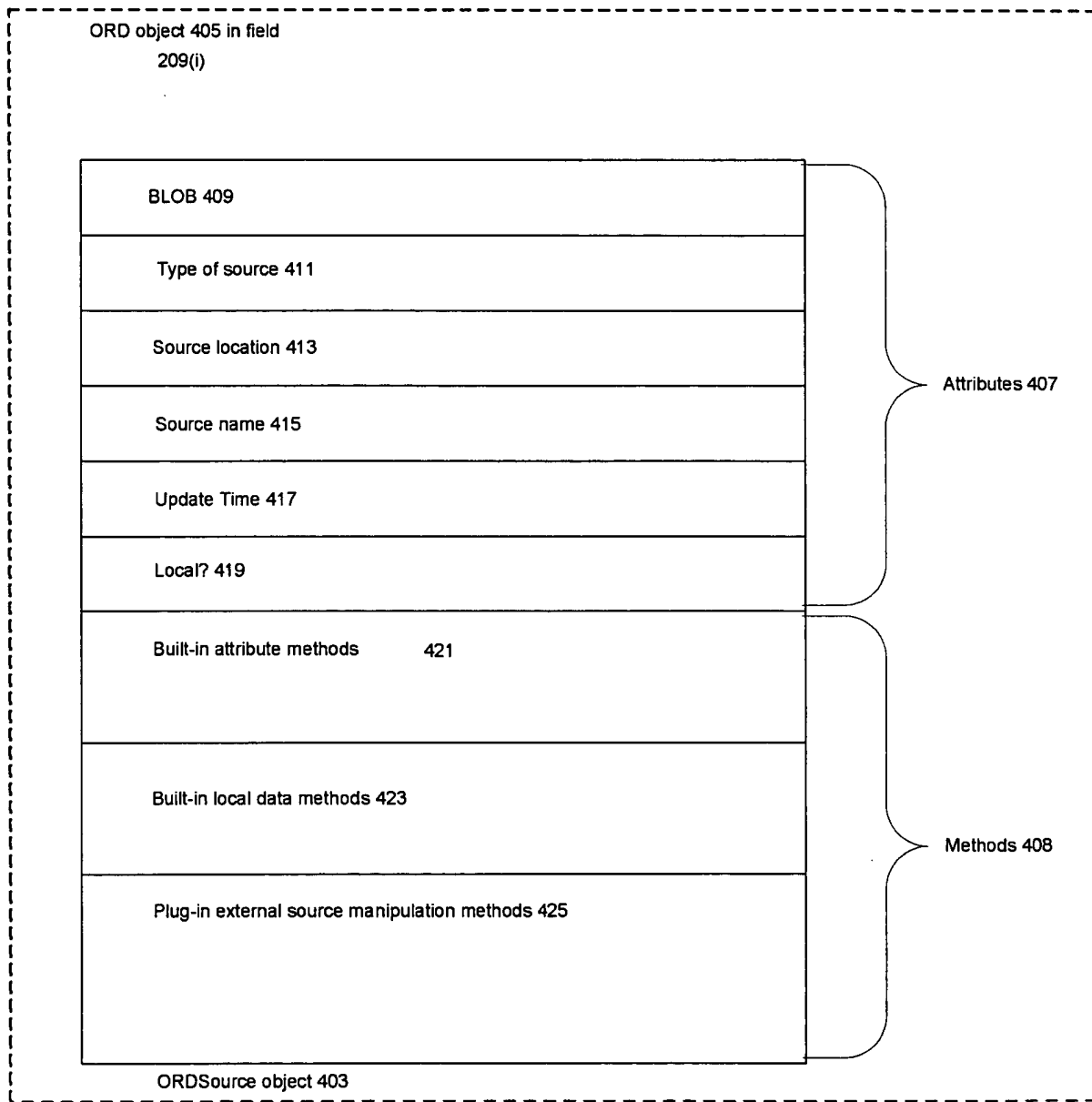
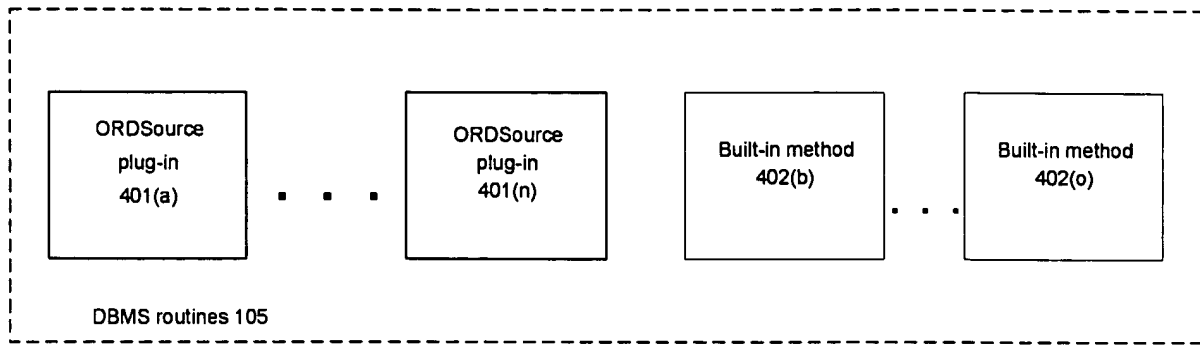


FIG. 4

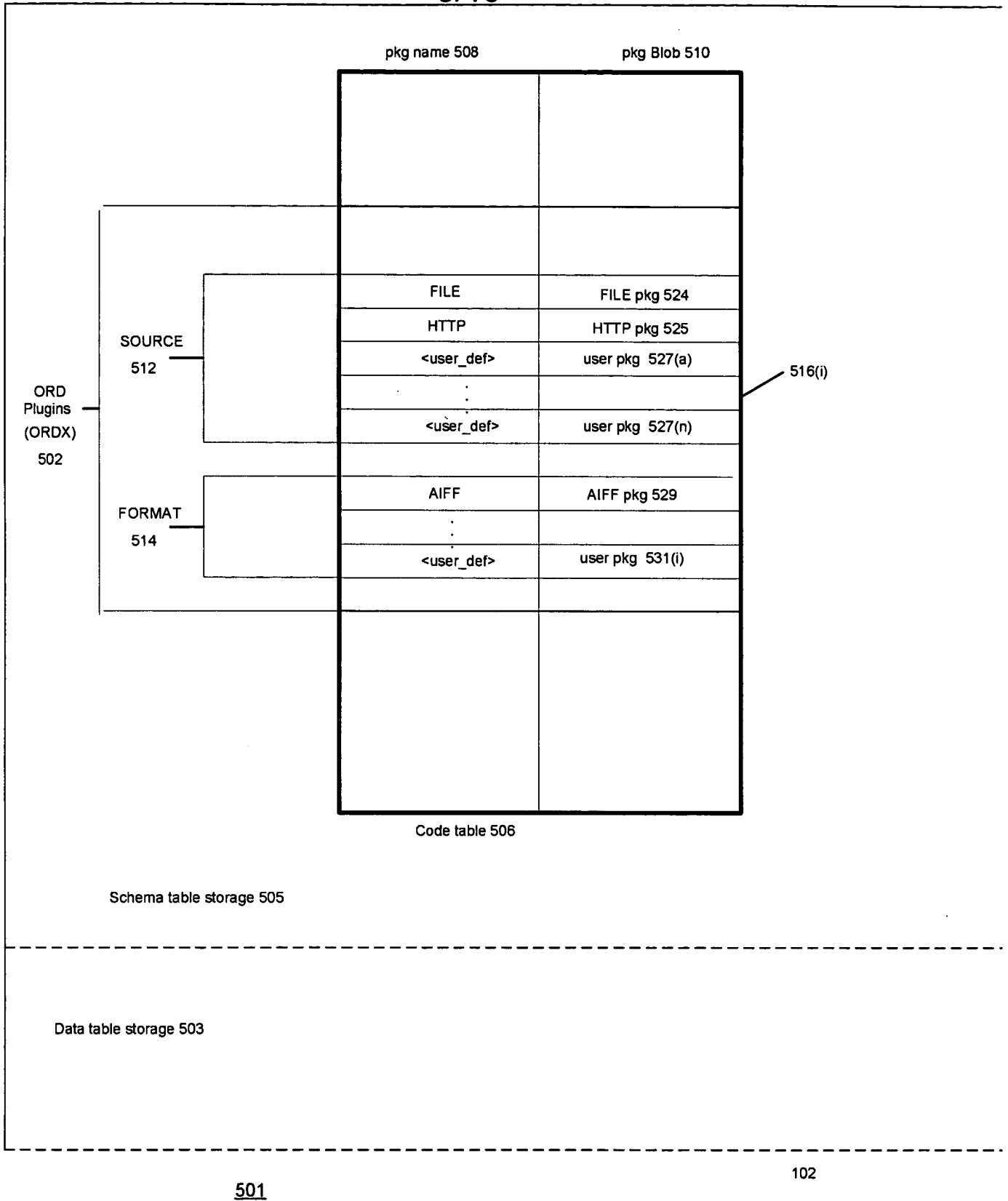



FIG. 5

6/16

```
610 CREATE OR REPLACE PACKAGE BODY ORDX_MY_SOURCE
    AS
        -- functions/procedures
        611 FUNCTION processCommand(
                obj IN OUT NOCOPY ORDSYS.ORDSource,
                ctx IN OUT RAW,
                cmd IN VARCHAR2,
                arglist IN VARCHAR2 605
                result OUT RAW)
        RETURN RAW
        IS
            --Your variables go here 607
            BEGIN
                --Your code goes here 609
            END processCommand;
        612 PROCEDURE import( obj IN OUT NOCOPY ORDSYS.ORDSource,
                ctx IN OUT RAW,
                dlob IN OUT NOCOPY BLOB,
                mimetype OUT VARCHAR2,
                format OUT VARCHAR2)
        IS
            --Your variables go here
            BEGIN
                --Your code goes here
            END import;
        613 PROCEDURE importFrom( obj IN OUT NOCOPY
                ORDSYS.ORDSource,
                ctx IN OUT RAW,
                dlob IN OUT NOCOPY BLOB,
                mimetype OUT VARCHAR2,
                format OUT VARCHAR2
                loc IN VARCHAR2,
                name IN VARCHAR2)
        IS
            --Your variables go here
            BEGIN
                --Your code goes here
            END importFrom;
```



603

601

FIG. 6 A

```

615 PROCEDURE export( obj IN OUT NOCOPY ORDSYS.ORDSource,
                     ctx IN OUT RAW,
                     dlob IN OUT NOCOPY BLOB,
                     loc IN VARCHAR2,
                     name IN VARCHAR2)
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END export;
617 FUNCTION getContentLength( obj IN ORDSYS.ORDSource,
                             ctx IN OUT RAW)
    RETURN INTEGER
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END getContentLength;
619 FUNCTION getSourceAddress(obj IN ORDSYS.ORDSource,
                             ctx IN OUT RAW,
                             userData IN VARCHAR2)
    RETURN VARCHAR2
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END getSourceAddress;
621 FUNCTION open(obj IN OUT NOCOPY ORDSYS.ORDSource,
                 userArg IN RAW,
                 ctx OUT RAW)
    RETURN INTEGER
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END open;
623 FUNCTION close(obj IN OUT NOCOPY ORDSYS.ORDSource,
                  ctx IN OUT RAW)
    RETURN INTEGER
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END close;

```

FIG. 6 B

```

625 FUNCTION trim(obj IN OUT NOCOPY ORDSYS.ORDSource,
                ctx IN OUT RAW,
                newlen IN INTEGER)
    RETURN INTEGER
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END trim;
627 PROCEDURE read(obj IN OUT NOCOPY ORDSYS.ORDSource,
                ctx IN OUT RAW,
                startPos IN INTEGER,
                numBytes IN OUT INTEGER,
                buffer OUT RAW)
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END read;
629 PROCEDURE write(obj IN OUT NOCOPY ORDSYS.ORDSource,
                ctx IN OUT RAW,
                startPos IN INTEGER,
                numBytes IN OUT INTEGER,
                buffer OUT RAW)
    IS
    --Your variables go here
    BEGIN
    --Your code goes here
    END write;
END ORDX_MY_SOURCE;
/
show errors;

```

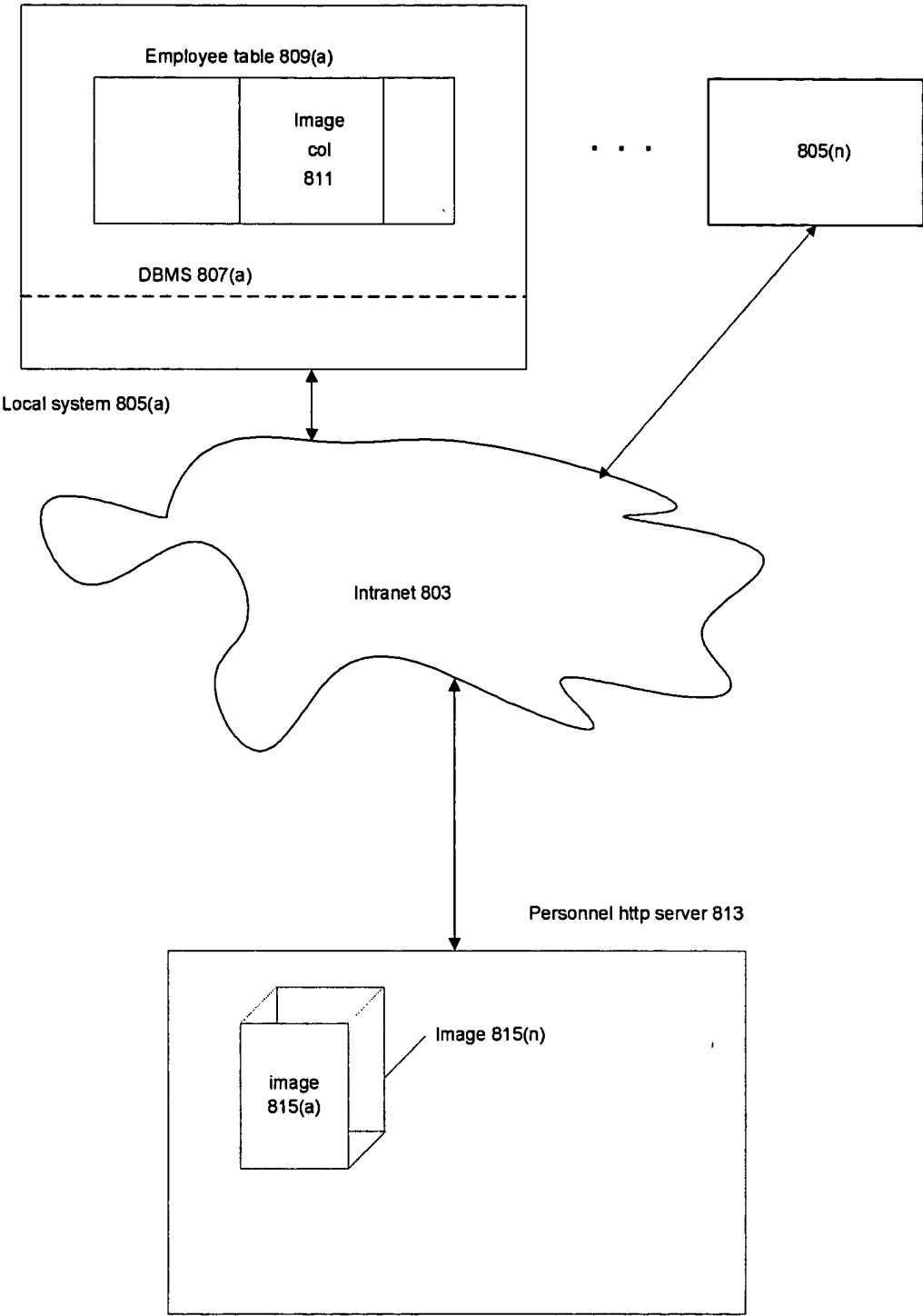
FIG. 6 C

9/16

```
DECLARE
  obj ORDSYS.ORDSource ;
  res RAW(4000);
  result RAW(4000);
  command VARCHAR2(4000);
  argList VARCHAR2(4000);
  ctx RAW(4000) :=NULL;
BEGIN
  select s into obj from TS where N =1 for UPDATE; 705
  command := 'xxx ';
  argList := 'yyy ';
  res := obj.processCommand(ctx, command, argList, result); 709
  UPDATE TS SET s=obj WHERE N=1 ; 711
  COMMIT;
  EXCEPTION
    WHEN ORDSYS.ORDSourceExceptions.METHOD_NOT_SUPPORTED THEN
      DBMS_OUTPUT.put_line('Source METHOD NOT SUPPORTED aught');
    WHEN ORDSYS.ORDSourceExceptions.SOURCE_PLUGIN EXCEPTION THEN
      DBMS_OUTPUT.put_line('SOURCE PLUGIN EXCEPTION caught');
    WHEN OTHERS THEN
      DBMS_OUTPUT.put_line('OTHER EXCEPTION caught');
END;
/
701
```

The diagram consists of several curly braces and line numbers placed to the left and right of the code. A large brace on the right groups the first five lines of the DECLARE block, with the number 703 to its right. A brace on the right groups the two lines of the select statement, with the number 705 to its right. A brace on the right groups the two lines of the update statement, with the number 707 to its right. A brace on the right groups the two lines of the processCommand call, with the number 709 to its right. A brace on the right groups the two lines of the update statement, with the number 711 to its right. A brace on the left groups the three lines of the exception block, with the number 713 to its left.

FIG. 7



801

FIG. 8

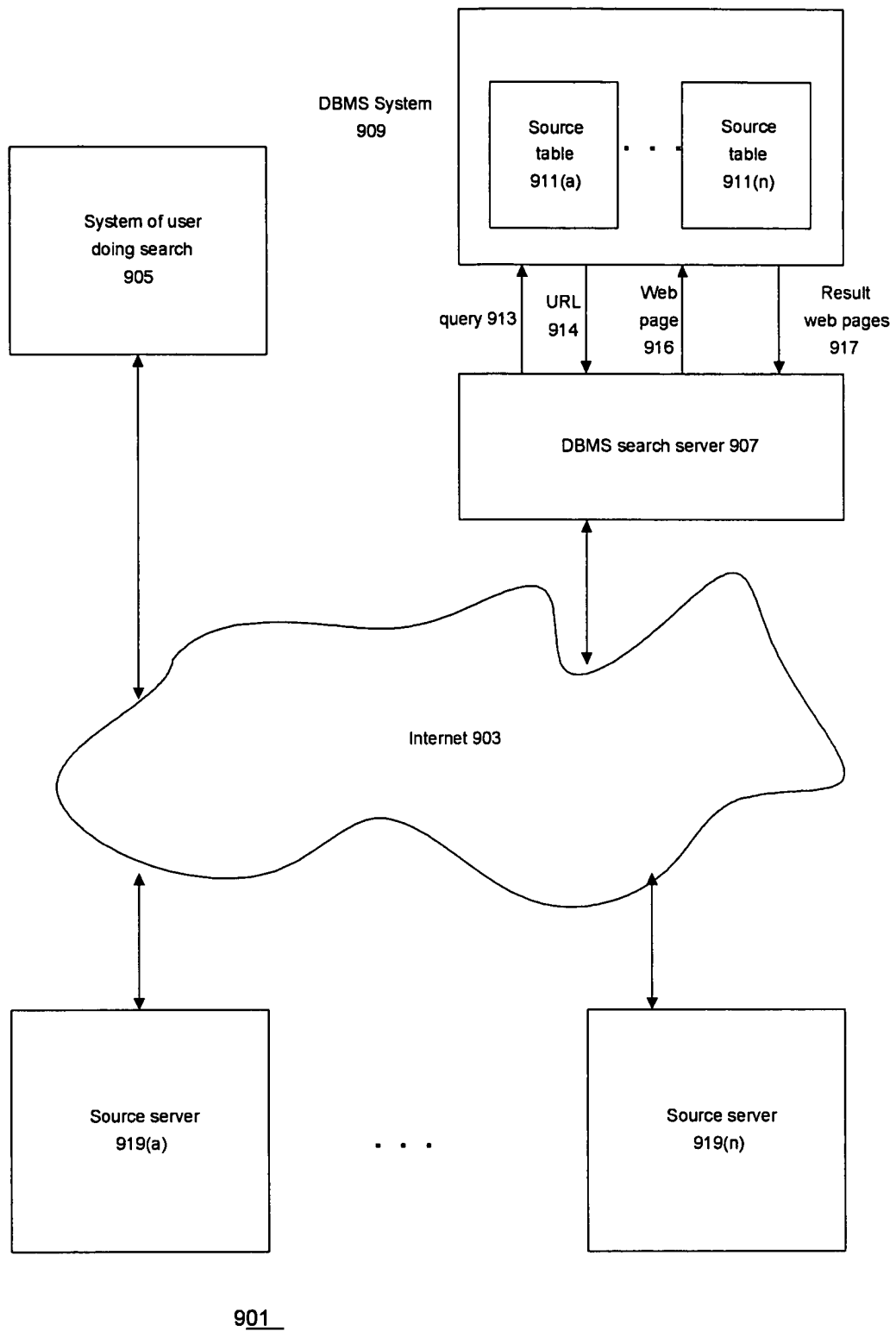
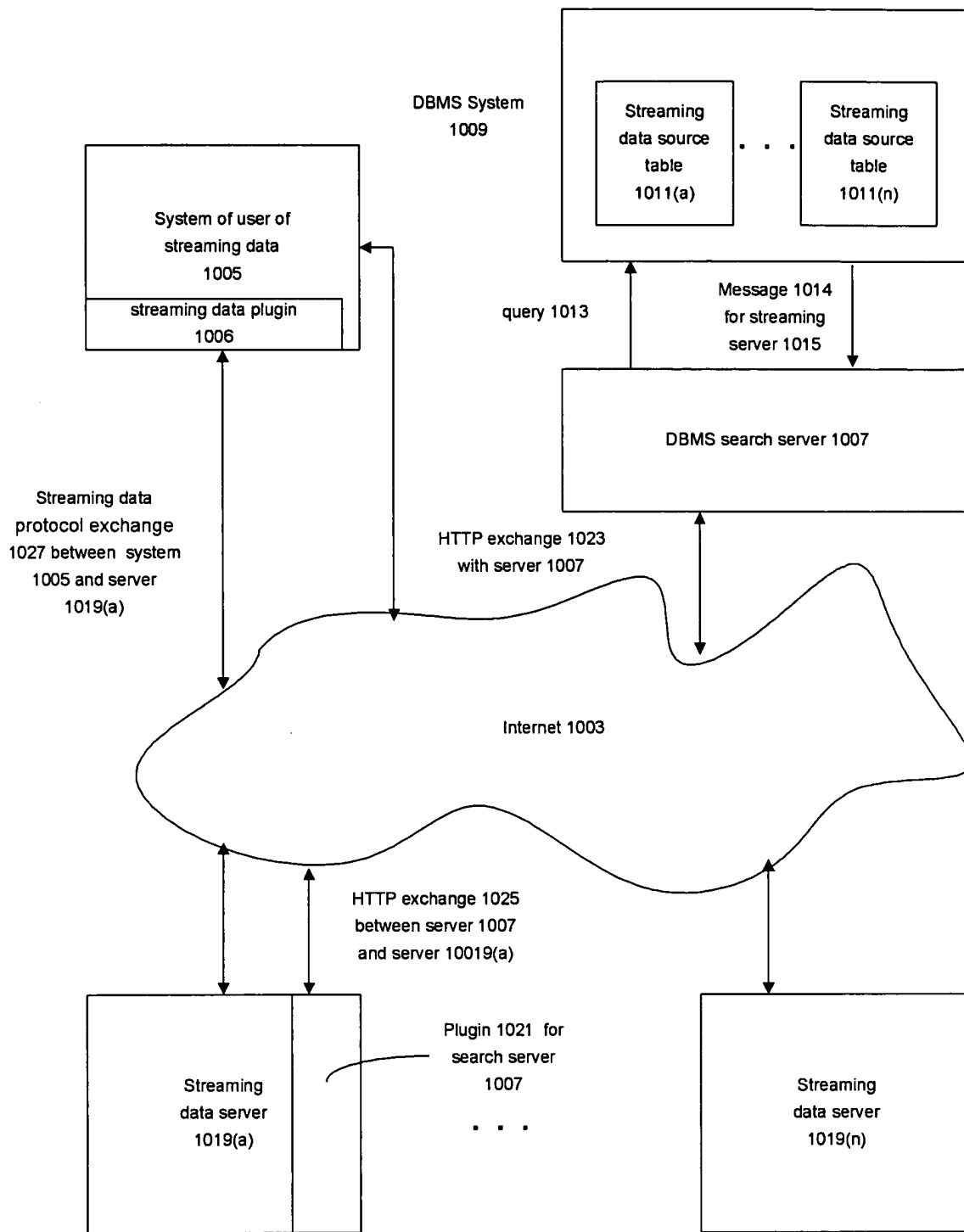


FIG. 9



1001

FIG. 10

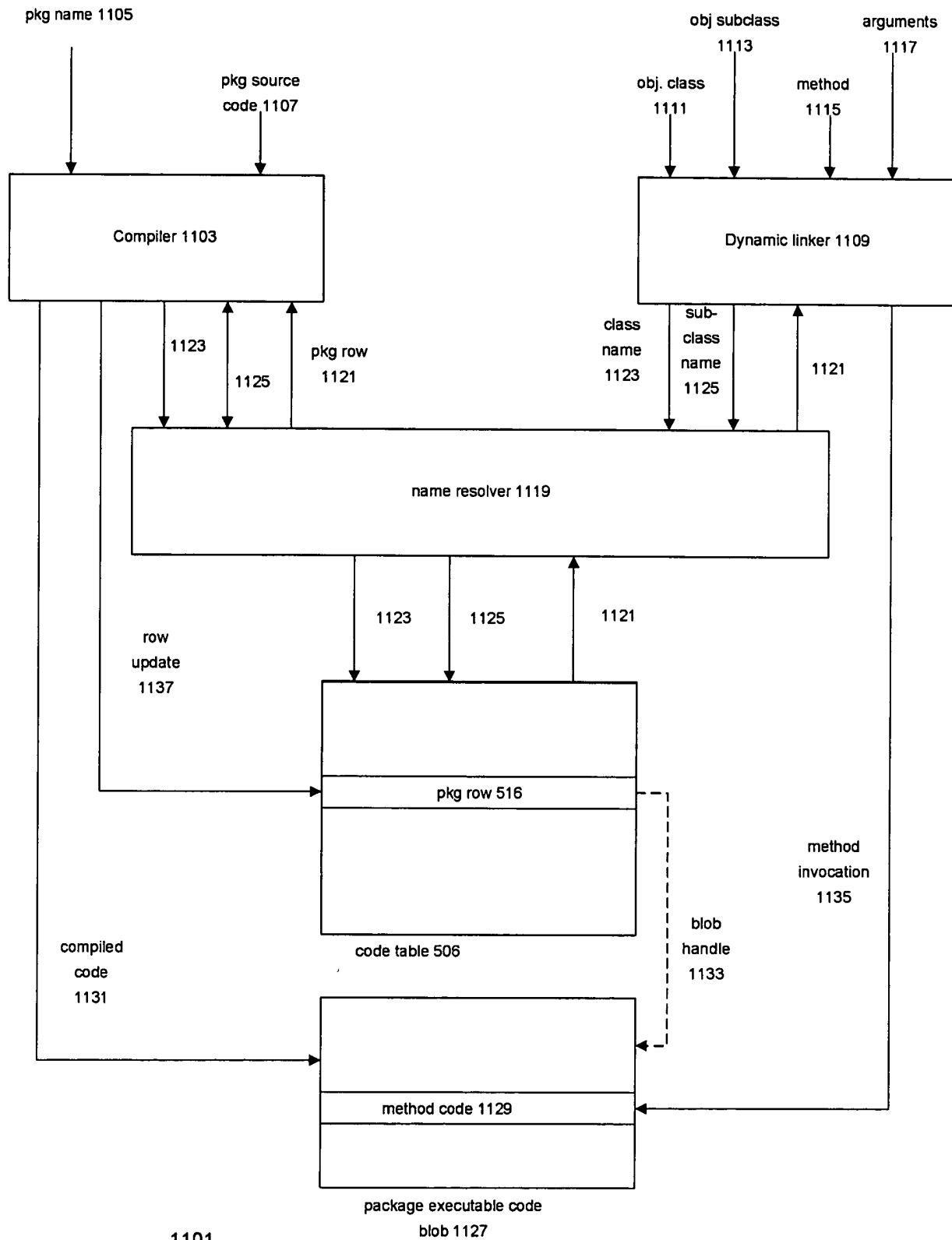


FIG. 11

14/16

```
CREATE OR REPLACE TYPE ORDAudio
AS OBJECT
(
  -- ATTRIBUTES 1203

  description VARCHAR2(4000),
  source ORDSOURCE,
  format VARCHAR2(31), 1205
  mimeType VARCHAR2(4000),
  comments CLOB,
  -- AUDIO RELATED ATTRIBUTES 1207
  encoding VARCHAR2(256),
  numberOfChannels INTEGER,
  samplingRate INTEGER,
  sampleSize INTEGER,
  compressionType VARCHAR2(4000),
  audioDuration INTEGER,

  -- METHODS 1209
  -- Methods associated with audio attributes accessors 1211
  MEMBER PROCEDURE setFormat(knownFormat IN VARCHAR2), 1215
  MEMBER FUNCTION getFormat RETURN VARCHAR2, 1217
  MEMBER FUNCTION getFormat(ctx IN OUT RAW) RETURN VARCHAR2, 1219
  MEMBER PROCEDURE setEncoding(knownEncoding IN VARCHAR2),
  MEMBER FUNCTION getEncoding RETURN VARCHAR2,
  MEMBER FUNCTION getEncoding(ctx IN OUT RAW) RETURN VARCHAR2,
  MEMBER PROCEDURE setNumberOfChannels(knownNumberOfChannels IN
    INTEGER),
  MEMBER FUNCTION getNumberOfChannels RETURN INTEGER,
  MEMBER FUNCTION getNumberOfChannels(ctx IN OUT RAW) RETURN
    INTEGER,
  MEMBER PROCEDURE setSamplingRate(knownSamplingRate IN
    INTEGER),
  MEMBER FUNCTION getSamplingRate RETURN INTEGER,
  MEMBER FUNCTION getSamplingRate(ctx IN OUT RAW) RETURN
    INTEGER,
  MEMBER PROCEDURE setSampleSize(knownSampleSize IN INTEGER),
  MEMBER FUNCTION getSampleSize RETURN INTEGER,
  MEMBER FUNCTION getSampleSize(ctx IN OUT RAW) RETURN INTEGER,
  MEMBER PROCEDURE setCompressionType(knownCompressionType IN
    VARCHAR2),
  MEMBER FUNCTION getCompressionType RETURN VARCHAR2,
  MEMBER FUNCTION getCompressionType(ctx IN OUT RAW) RETURN
    VARCHAR2,
  MEMBER PROCEDURE setAudioDuration(knownAudioDuration IN
    INTEGER),
  MEMBER FUNCTION getAudioDuration RETURN INTEGER,
  MEMBER FUNCTION getAudioDuration(ctx IN OUT RAW) RETURN
    INTEGER,

  1201

```

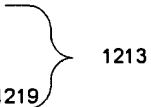


FIG. 12A

```

MEMBER PROCEDURE setKnownAttributes(
    knownFormat IN VARCHAR2,
    knownEncoding IN VARCHAR2,
    knownNumberOfChannels IN INTEGER,
    knownSamplingRate IN INTEGER,
    knownSampleSize IN INTEGER,
    knownCompressionType IN VARCHAR2,
    knownAudioDuration IN INTEGER),

-- Methods associated with setting all the properties
MEMBER PROCEDURE setProperties(ctx IN OUT RAW),
MEMBER FUNCTION checkProperties(ctx IN OUT RAW) RETURN
    BOOLEAN,
MEMBER FUNCTION getAttribute(
    ctx IN OUT RAW,
    name IN VARCHAR2) RETURN VARCHAR2,
MEMBER PROCEDURE getAllAttributes(
    ctx IN OUT RAW,
    attributes IN OUT NOCOPY CLOB),

-- Methods associated with audio processing 1221
MEMBER FUNCTION processCommand(
    ctx IN OUT RAW,
    cmd IN VARCHAR2,
    arguments IN VARCHAR2,
    result OUT RAW)
RETURN RAW

```

1201

FIG. 12B

```

CREATE OR REPLACE PACKAGE ORDX_AIFF_AUDIO 1303
authid current_user
AS

--AUDIO ATTRIBUTES ACCESSORS
FUNCTION getFormat(ctx IN OUT RAW, obj IN ORDSYS.ORDAudio) 1213
    RETURN VARCHAR2;
FUNCTION getEncoding(ctx IN OUT RAW, obj IN
    ORDSYS.ORDAudio)
    RETURN VARCHAR2;
FUNCTION getNumberOfChannels(ctx IN OUT RAW, obj IN
    ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getSamplingRate(ctx IN OUT RAW, obj IN
    ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getSampleSize(ctx IN OUT RAW, obj IN
    ORDSYS.ORDAudio)
    RETURN INTEGER;
FUNCTION getCompressionType(ctx IN OUT RAW, obj IN
    ORDSYS.ORDAudio)
    RETURN VARCHAR2;
FUNCTION getAudioDuration(ctx IN OUT RAW, obj IN
    ORDSYS.ORDAudio)
    RETURN INTEGER;
PROCEDURE setProperties(ctx IN OUT RAW, obj IN OUT NOCOPY
    ORDSYS.ORDAudio);
FUNCTION checkProperties(ctx IN OUT RAW, obj IN OUT
    ORDSYS.ORDAudio)
    RETURN NUMBER;
FUNCTION getAttribute(ctx IN OUT RAW,
    obj IN ORDSYS.ORDAudio,
    name IN VARCHAR2) RETURN VARCHAR2;
-- must return name=value; name=value; ... pairs
PROCEDURE getAllAttributes(ctx IN OUT RAW,
    obj IN ORDSYS.ORDAudio,
    attributes IN OUT NOCOPY CLOB);

-- AUDIO PROCESSING METHODS
FUNCTION processCommand(
    ctx          IN OUT RAW,
    obj          IN OUT NOCOPY
    ORDSYS.ORDAudio,
    cmd          IN VARCHAR2,
    arguments IN VARCHAR2,
    result OUT RAW)
    RETURN RAW;

END;
/
show errors;

```

1301

FIG. 13